Software User's Manual

Mission Science iRobots

Team 07

Jiashuo Li	**Project Manager, Life Cycle Planner, Developer**
Chen Li	**Requirements Engineer, Software Architect, Developer**
Farica Mascarenhas	**Operational Concept Engineer, IV&V, Quality Analyst**
Hanadi Mardah	**Tester**
Sergey Mukhin	**Prototyper, Developer**
Yun Shao	**Feasibility Analyst, Developer**

Version History

Date	Author	Version	Changes made	Rationale
03/25/15	CL	1.0	• Add Introduction and Operational Procedures	• For CCD

Table of Contents

VERSION HISTORY ..... **II**

TABLE OF CONTENTS ..... **III**

TABLE OF TABLES ..... **IV**

TABLE OF FIGURES ..... **V**

1. Introduction 1

    1.1 System Overview 1

    1.2 System Requirements 1

2. Operational Procedures 2

    2.1 System Menu 2

    2.2 Instructions 5

    2.3 File 11

    2.4 Build 12

Table of Tables

No table of figures entries found.

Table of Figures

Figure 1 System Menu 2

Figure 2 Showing Description 3

Figure 3 Reorder 4

Figure 4 Source Code Generated 4

Figure 5 Move 5

Figure 6 Rotate 6

Figure 7 Drive 7

Figure 8 LED 8

Figure 9 Song 9

Figure 10 If Example 10

Figure 11 Loop Example 11

Figure 12 File 12

Figure 13 Build 13

Figure 14 WinAVR Configuration 13

1. Introduction

1.1 System Overview

Mission Science iRobots is a drag-and-drop graphical user interface, which allows users to control the behavior of an iRobot. The system aims to inspire elementary school kids to learn the logic and control systems and generate higher funds for Mission Science.

1.2 System Requirements

1.2.1 Hardware Requirements

None

1.2.2 Software Requirements

Windows 7 or above

1.2.3 Other Requirements

None

2. Operational Procedures

2.1 System Menu

The system contains 3 main sections: Instruction Panel, Program List and Source Code. An instruction can be dragged from Instruction Panel to Program List. Instruction can also be added to Program List by clicking “Load Example Code” on the right-hand side.

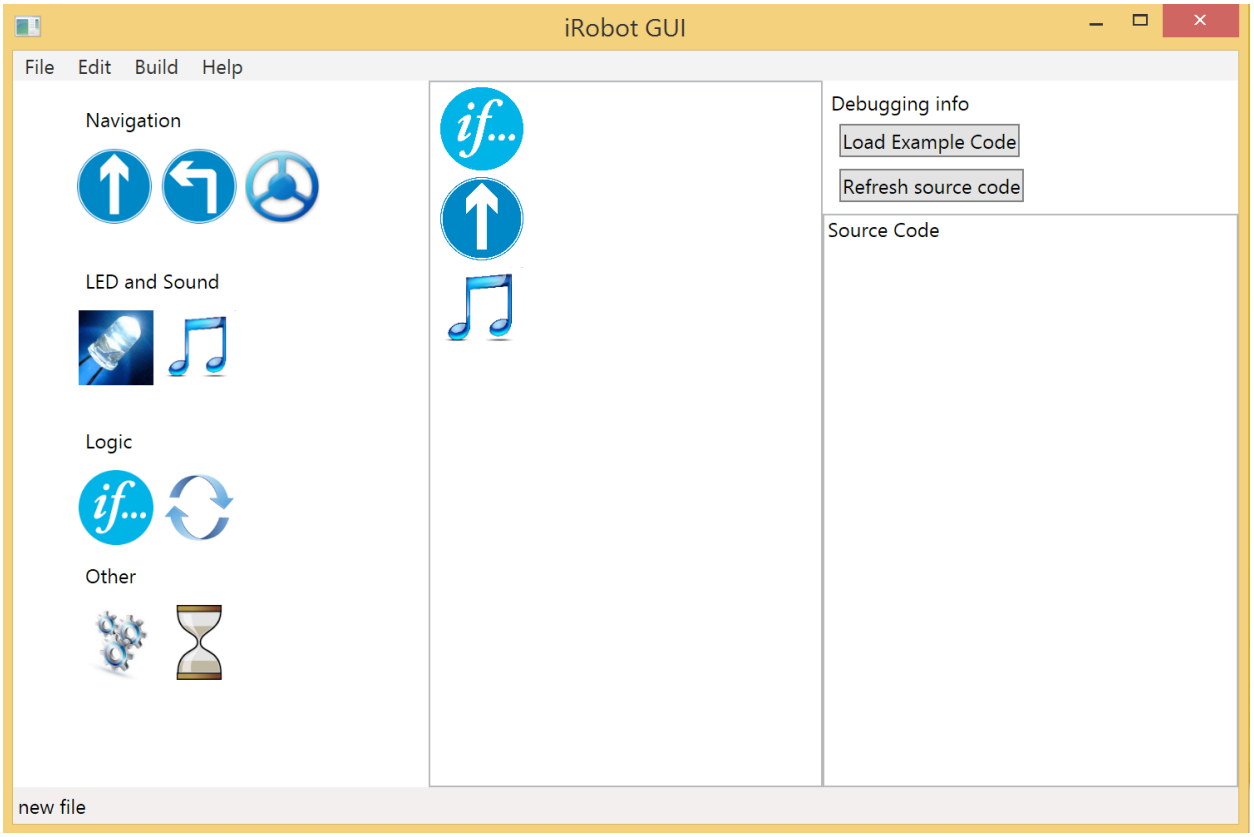

Figure 1 System Menu

2.1.1 Instruction Panel

In the Instruction Panel, there are 9 different instructions grouped in 4 sections.

- The “Navigation” section contains “Move”, “Rotate”, and “Drive”.
- The “LED and Sound” section contains “Turn on/off the LED” and “Define a song”.
- The “Logic” section contains “If” and “Loop”.
- The “Other” section contains “Play a demo” and “Delay”.

Each icon represents a specific instruction. If a user cannot tell which instruction the icon stands for, then he/she can pause the mouse pointer over an icon and a description will pop up with the name of the corresponding instruction.

Figure 2 Showing Description

2.1.2 Program List

- Change Parameters

  When an instruction in the Program List is double clicked, then a new window will pop up so that parameters can be adjusted or new instructions can be added.

- Reorder

  Instructions can also be dragged and dropped inside Program List, as Figure 3 shows.

- Remove

  Pressing “Delete” key on the keyboard will remove the selected instruction.

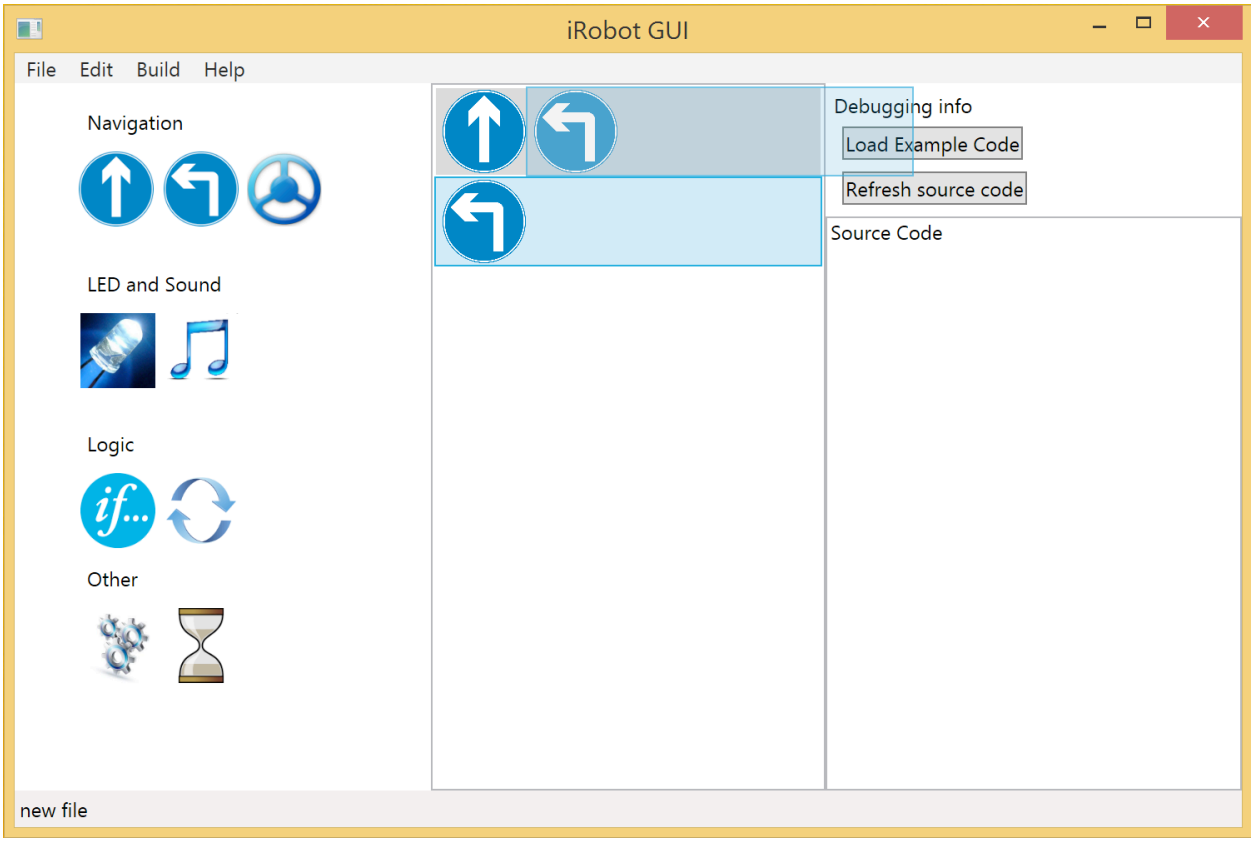

Figure 3 Reorder

2.1.3 Source Code

Once the "Refresh source code" button is clicked, the source code will be generated, according to the current instructions in the Program List. A user can use the source code anytime he/she wants to see what they have done or to find out the bugs inside the program.

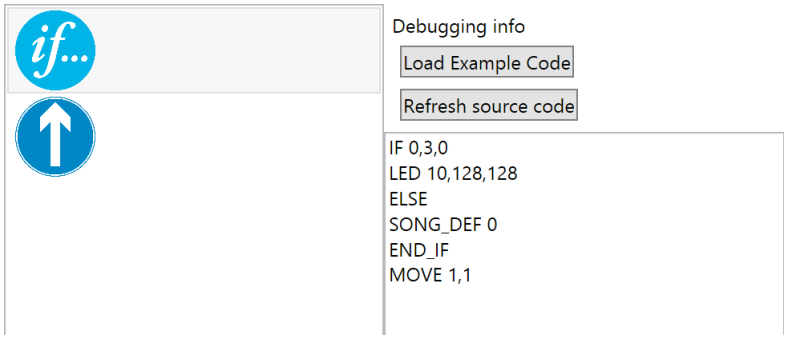

Figure 4 Source Code Generated

2.2 Instructions

2.2.1 Move

For the Move instruction, distance and duration can be changed in the popped up window. A user can drag the icon up and down to reset the value. The exact value is shown under the scroll bar.

- If distance is positive, then the robot will move forward; otherwise, it will move backward.
  - Range: -5000 ~ 5000mm
  - Default value: 300mm (forward)

- Duration means how long the robot will move.
  - Range: 0 ~ 5s
  - Default value: 3s

By setting both distance and duration, the velocity will be updated accordingly, since velocity = distance / duration.

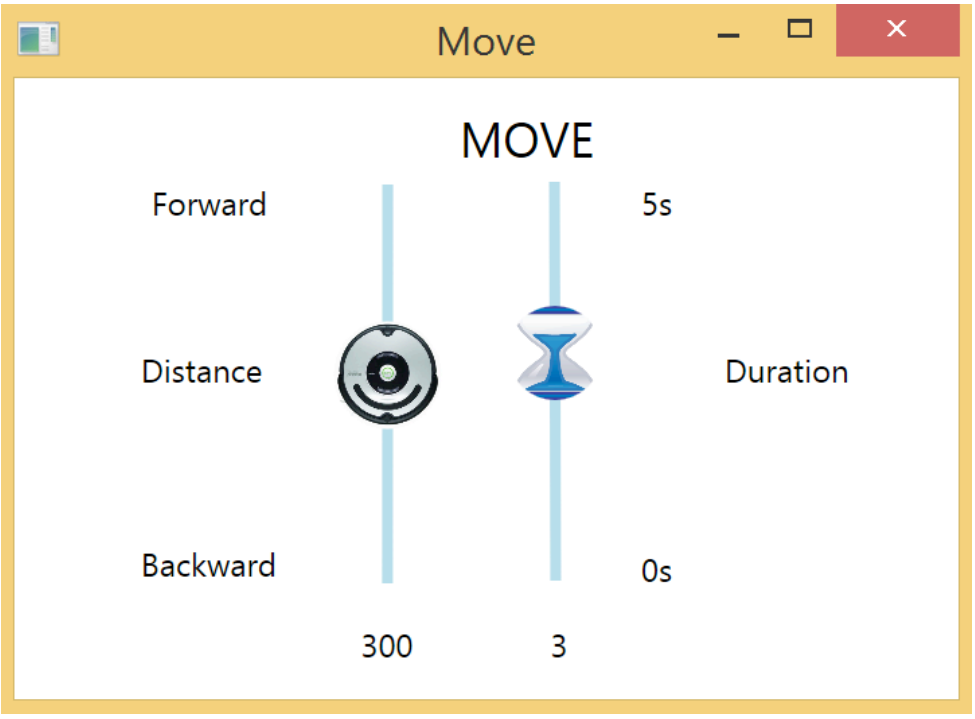

Figure 5 Move

2.2.2 Rotate

For the rotation instruction, there is a virtual iRobot in the newly opened window. By rotating the virtual robot, a user can specify in which direction and how much the real iRobot will rotate. The robot can rotate up to 90 degrees in both left and right directions.

Figure 6 Rotate

2.2.3 Drive

The drive instruction controls iRobot's drive wheels.

- Velocity is the average velocity of the drive wheels in millimeters per second (mm/s).
- Radius can also be specified, in millimeters, at which Roomba should turn. The longer radii make Roomba drive straighter; shorter radii make it turn more.

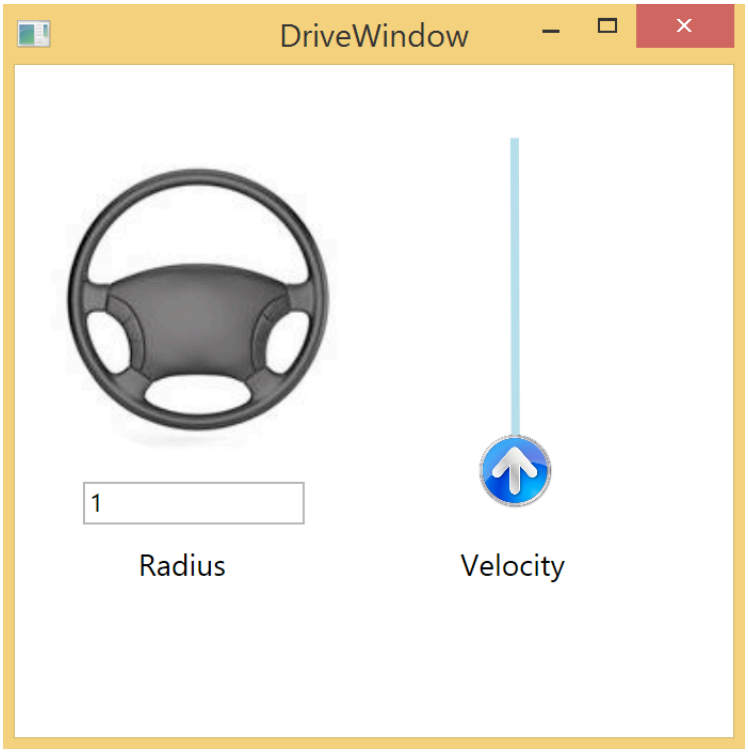

Figure 7 Drive

2.2.4 Turn on/off the LED

The LED instruction, by its name, controls the LEDs.

- There are 3 LEDs on the robot: Power LED, Play LED and Advance LED. A user can choose whether to turn on the last two LEDs by checking the box.
- Color is to control the color of Power LED. Color of Play and Advance LEDs cannot be changed. The color is changed as the color scroll bar is being dragged.
- Intensity is to change the intensity of Power LED. Intensity of Play and Advance LEDs cannot be changed. Intensity can be set to 0 to turn off the Power LED.

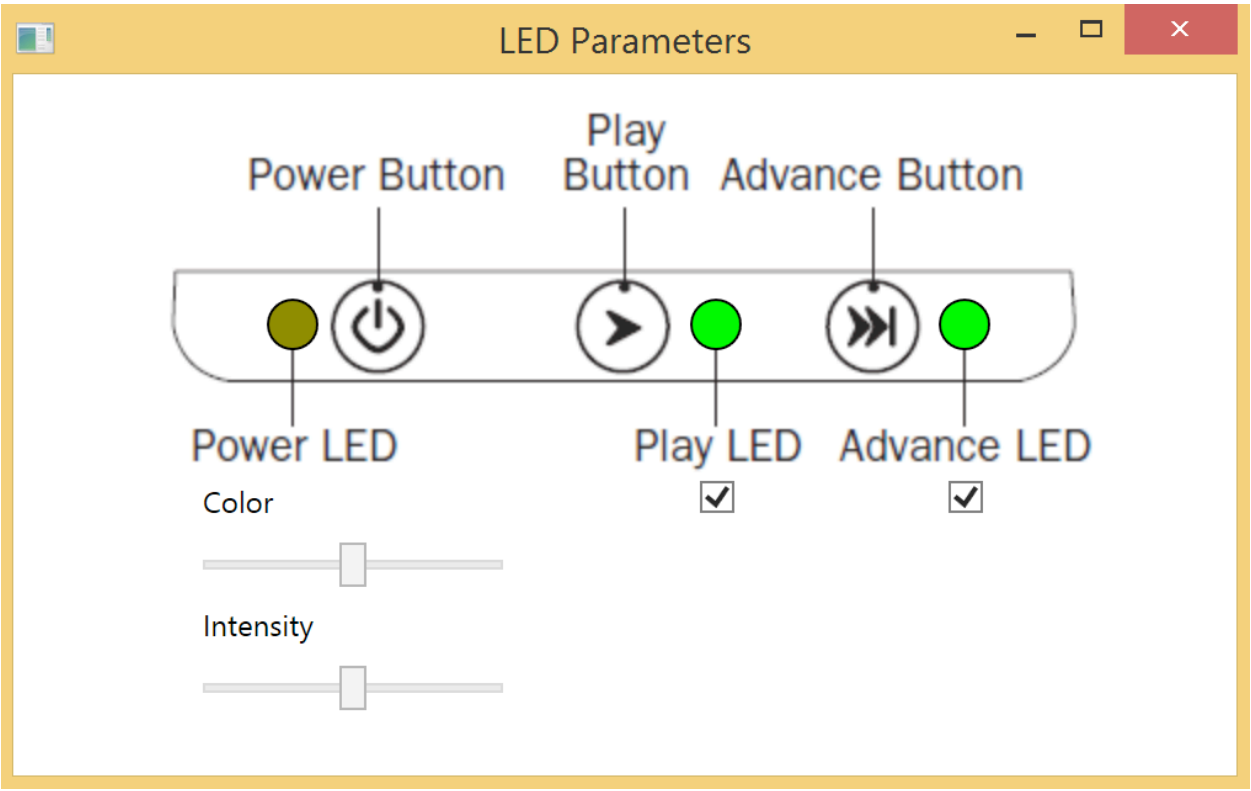

Figure 8 LED

2.2.5 Define a song

In the Song window, a user can create a song within 16 notes. The song will play automatically on the robot.

- New: clear the current contents and create a new song
- Up: swap the current note with the upper one
- Down: swap the current note with the lower one
- Cross: remove the selected note
- Scroll bar: change the volume

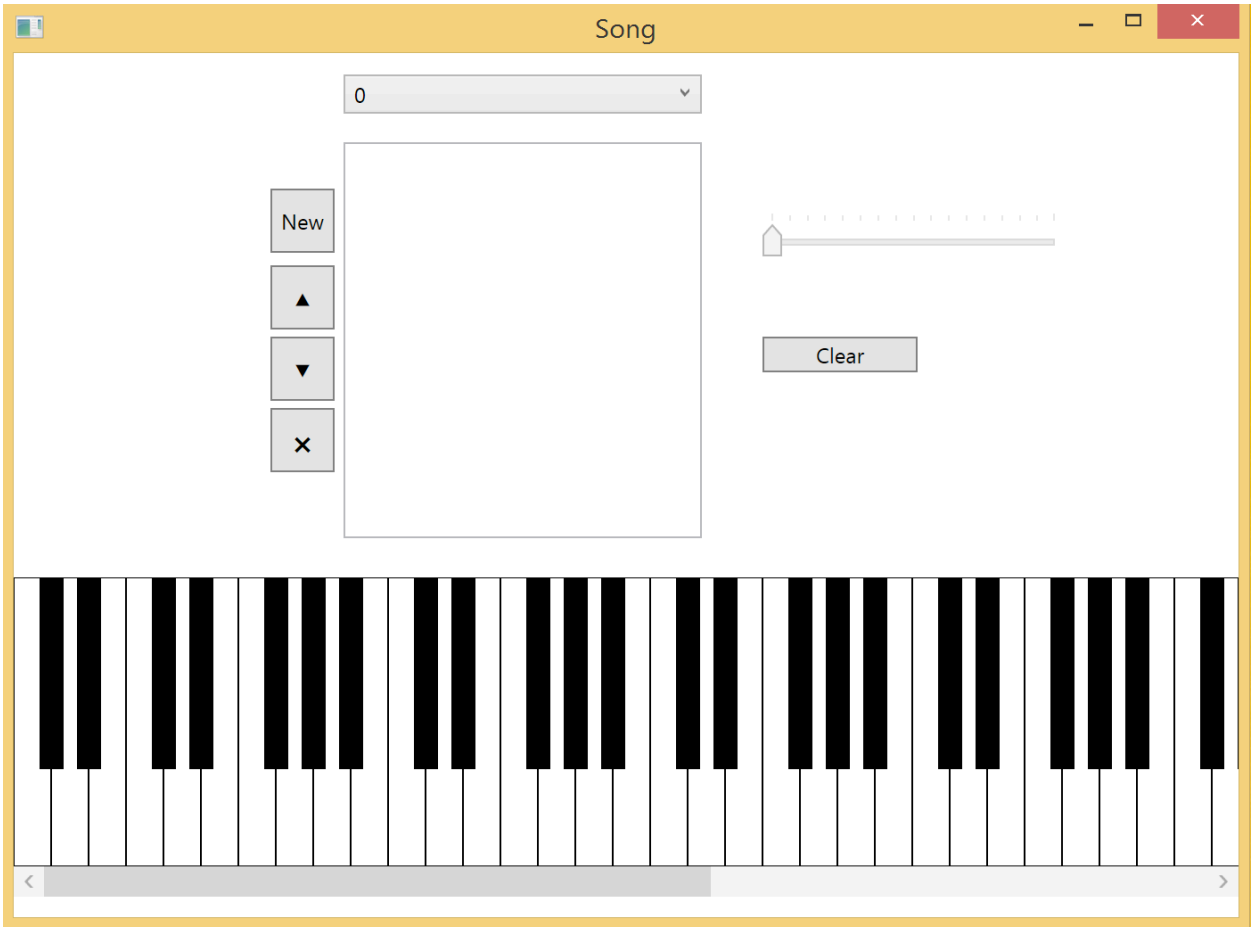

Figure 9 Song

2.2.6 If

The if window contains 3 sections: condition, true and false. If the condition is true, the instructions in the True panel are executed. Otherwise, the execution continues in the following branch – either in the False panel, or if there is no else branch, then it will do nothing.

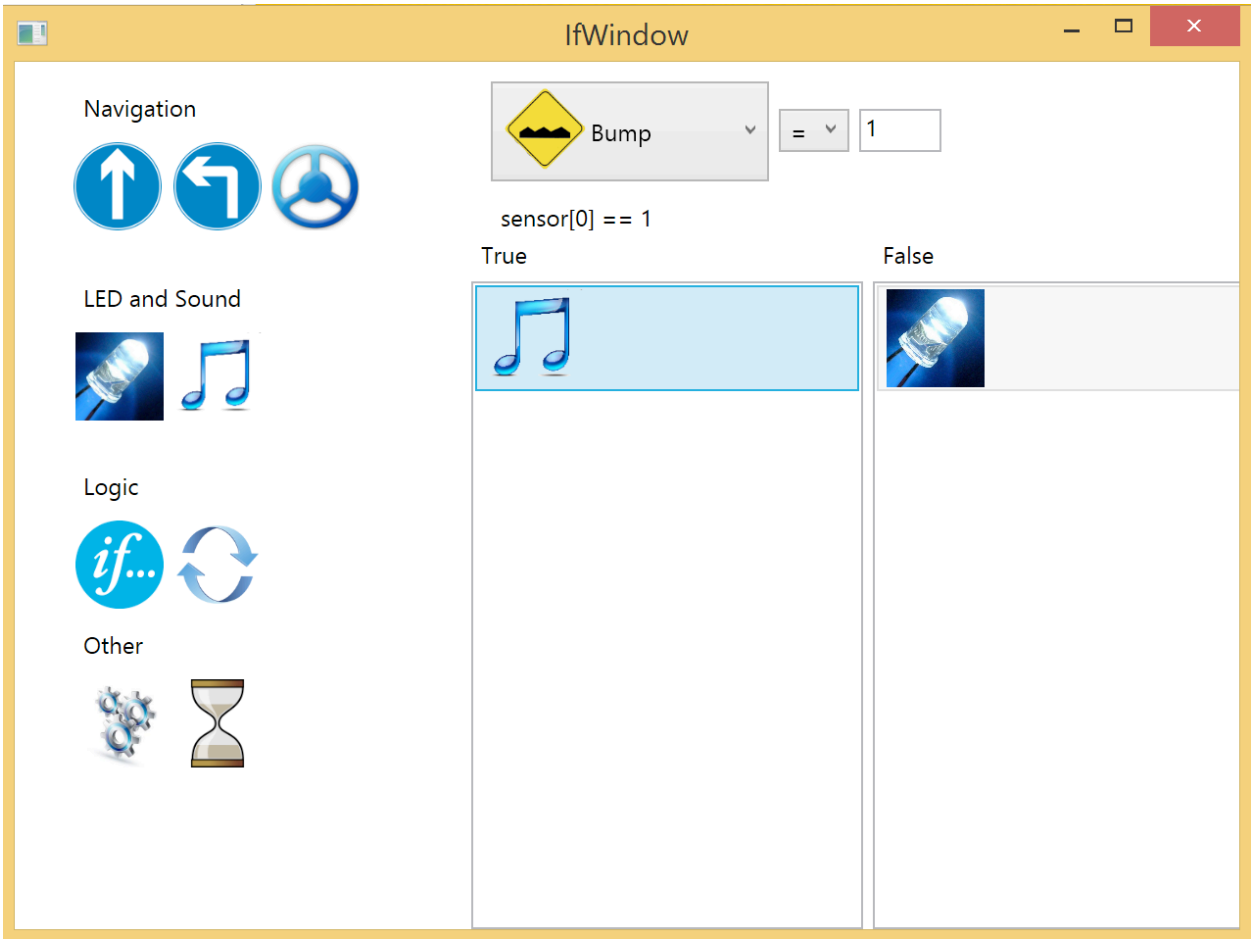

Figure 10 If Example

In this example, if the value of bump is not equal to 1, then the LED is turned on as user sets. When the robot meets the bump (value of bump = 1), the robot will start to play the song.

2.2.7 Loop

Loop is repeating some instructions under certain condition. The iRobot will repeat the loop until the condition changes.

The loop windows looks a little similar to the if window, both of which has the condition panel. What's different from the if window is that the loop window only has 1 Program List.

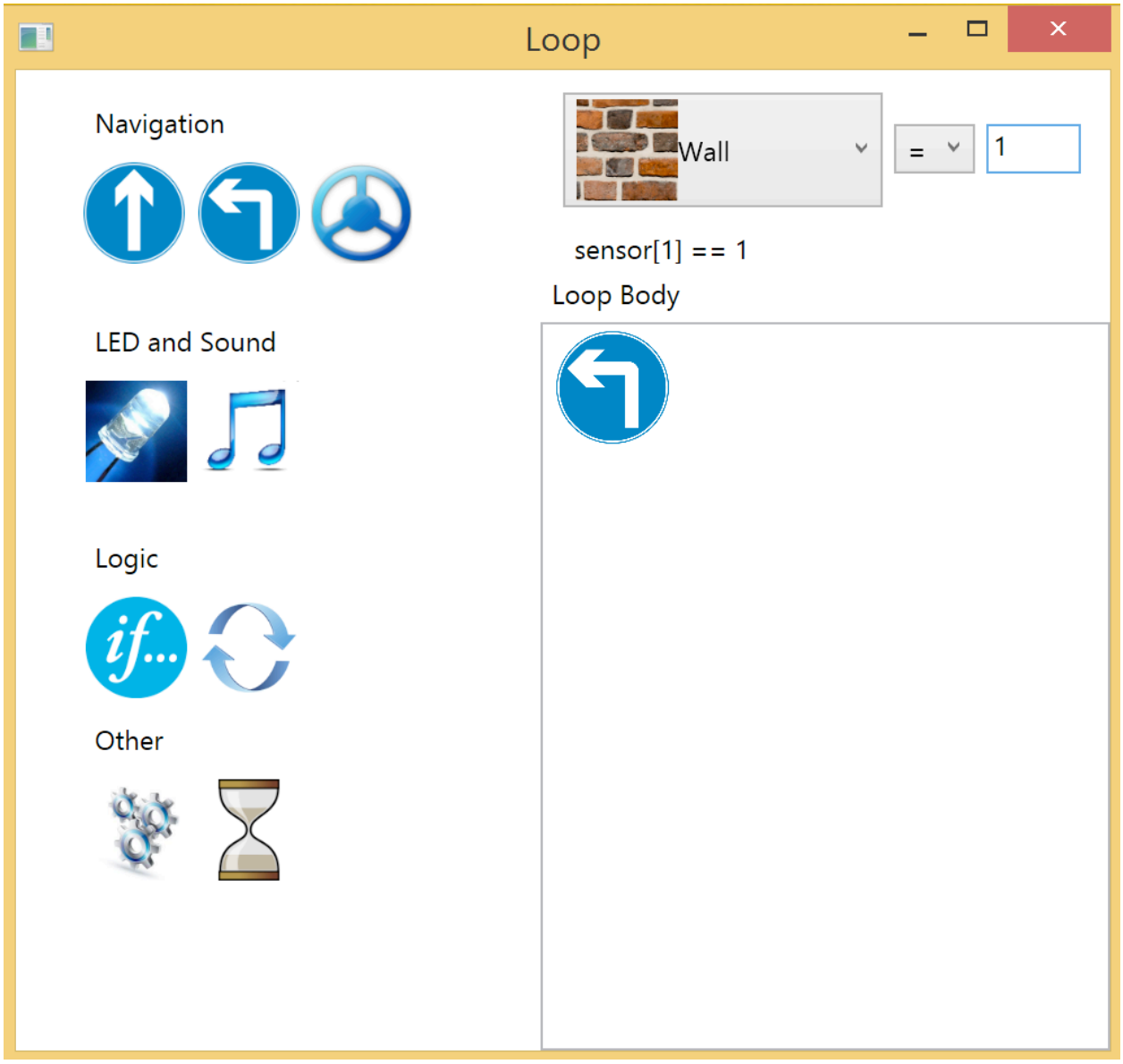

Figure 11 Loop Example

In this example, if the robot meets a wall (wall = 1), then it will keep rotating until it leaves the wall.

2.3 File

- New: a new program will be created without saving current work
- Save: save the file in igp format for future use or study

- Open: open a previously saved file

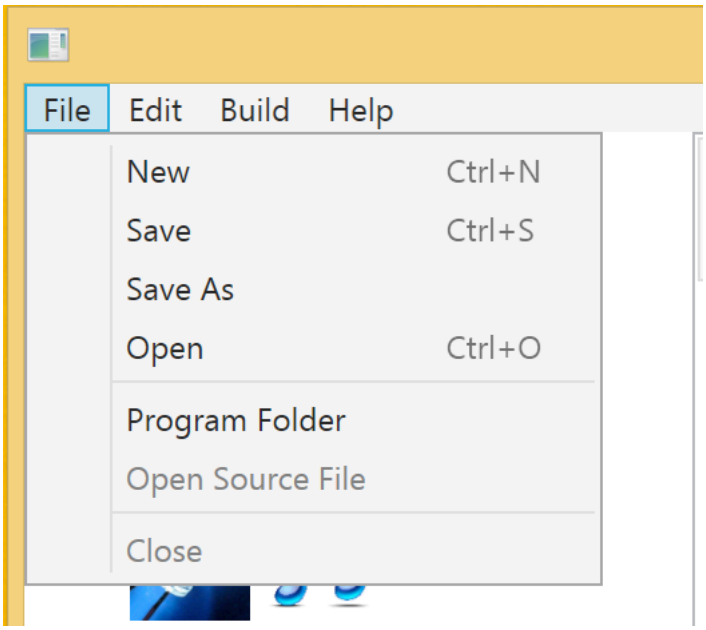

Figure 12 File

2.4 Build

- Translate to C Code: translate the source code into C Code, which the WinAVR will use.
- Show C Code: show the C Code just translated
- WinAVR
  - Build: build the program
  - Load: load the program to Microcontroller
  - Clean: clean the program inside Microcontroller which was previously loaded
- Load to Microcontroller: all-in-one function, translates the source code, builds the program and loads to Microcontroller
- WinAVR Configuration: COM Port needs configuration before program is loaded. To configure the COM Port, first click the button “Device Manager”, and then find the item “Ports (COM & LPT)”.
- Settings: provide convenience for future developers to maintain the system

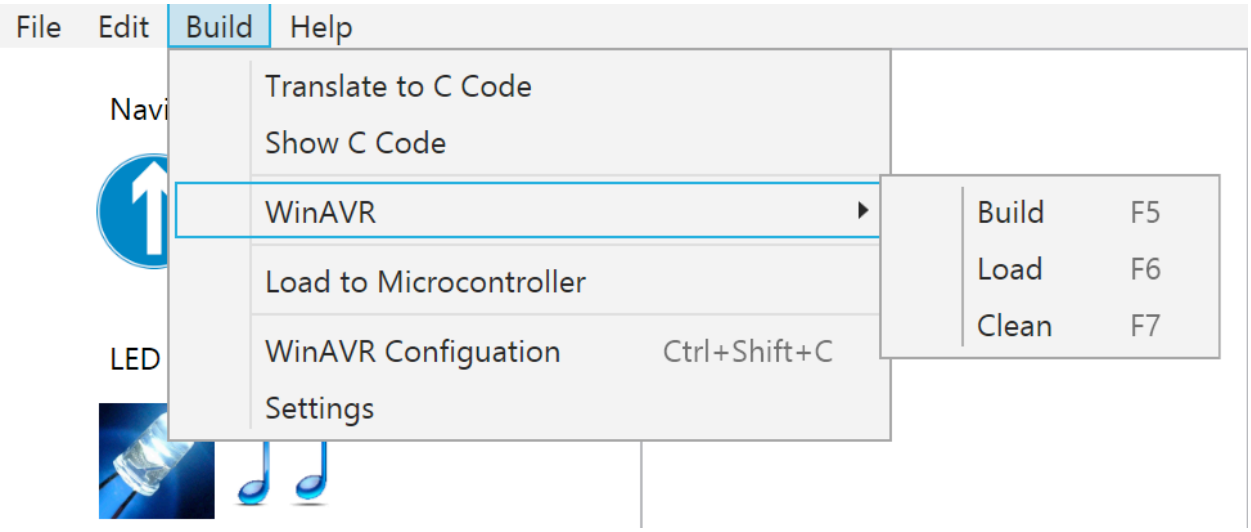

Figure 13 Build

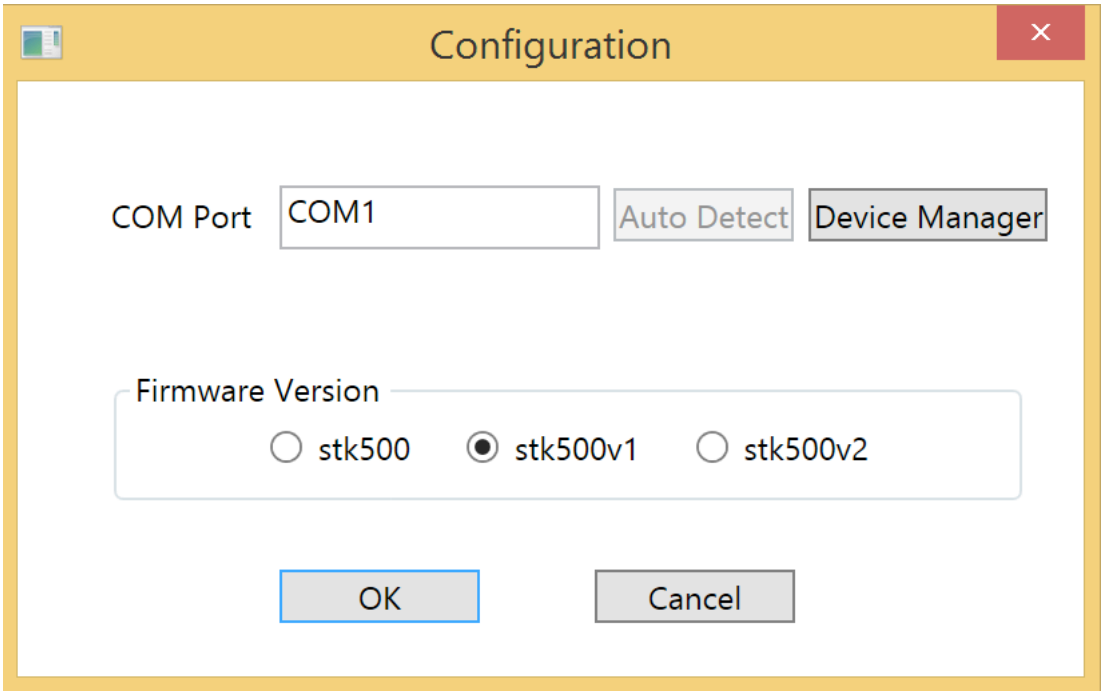

Figure 14 WinAVR Configuration